AD-A094 808   BOLT BERANEK AND NEWMAN INC CAMBRIDGE MA   F/G 5/9
               COMPUTER-BASED INTERACTIVE FLIGHT TRAINING.(U)
               NOV 80   W FEURZEIG, G LUKAS, D STEFANESCU   F49620-78-C-0066
UNCLASSIFIED   BBN-4530   AFOSR-TR-81-0088   NL

END
DATE
FILMED
3-81
DTIC

LEVEL II

(12)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS
BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
AFOSR-TR- 81-0088	AD-A094808	

4. TITLE (and Subtitle)	5. TYPE OF REPORT & PERIOD COVERED
COMPUTER-BASED INTERACTIVE FLIGHT TRAINING	Final April 1977–August 1930
	6. PERFORMING ORG. REPORT NUMBER 4530

7. AUTHOR(s)	8. CONTRACT OR GRANT NUMBER(s)
Wallace Feurzeig George Lukas Dan Stefanescu	F49620-78-C-0066

9. PERFORMING ORGANIZATION NAME AND ADDRESS	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
Bolt Beranek and Newman Inc. 50 Moulton St. Cambridge, MA 02138	61102F 2313A2

11. CONTROLLING OFFICE NAME AND ADDRESS	12. REPORT DATE
Air Force Office of Scientific Research Building 410 Bolling AFB, D. C. 20332 (NL)	November 1980
	13. NUMBER OF PAGES 42

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)	15. SECURITY CLASS. (of this report)
	Unclassified
	15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

DTIC
SELECTE
FEB 10 1981
E

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Adaptive training
Computer-aided instruction
Flight instruction
Performance analysis
Simulation

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Computer-based methods were developed for describing the salient features of pilot performance on complex flight tasks. These methods extend and generalize a new approach to performance analysis expressly designed to mirror the procedures used by expert instructors in analyzing simulator flight data and diagnosing trainee errors. A data base for guiding the development of this instructor model of performance analysis was provided by a set of ILS approach tasks flown on the ORLY simulator

DD FORM 1473 EDITION OF 1 NOV 65 IS OBSOLETE
1 JAN 73

AD A094808

DTIC FILE COPY

Final Report No. F49620-78-C-0066

COMPUTER-BASED INTERACTIVE FLIGHT TRAINING,

Bolt Beranek and Newman Inc.
50 Moulton Street
Cambridge, MA 02138

November 1980

Final Report, April 1977 — August 1980

Dod Distribution Statement

Approved for public release;
distribution unlimited.

Prepared for:

Life Sciences Directorate
Air Force Office of Scientific Research
Bolling AFB, D. C. 20332

Attn: Major Jack Thorpe

81 2 09 055

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

SECTION I

INTRODUCTION

This report describes the work performed under Contract No. F49620-78-C-0066 entitled "Computer-Based Interactive Flight Training." This is the final project report, covering the entire contract period 15 April 1978 through 15 August 1980. The contract was with the Air Force Office of Scientific Research. Major Jack Thorpe of the Life Sciences Directorate served as technical monitor.

The overall objective of the effort was the development of computer-based methods for analyzing and diagnosing pilot performance on complex flight tasks. In part'cular, the effort was aimed at extending and generalizing a new approach to performance analysis expressly designed to identify salient features and canonical errors in pilot performance on instrument flight tasks. The approach used in this work had been developed very recently. Its initial application was in the analysis of simple flight patterns such as clover leafs and figure 8s in basic instrument flight work. In the work reported here our research goal was to transform this earlier task-specific system into a general performance analysis system having a wide scope of application across flight tasks.

Our approach is modeled on the procedures used by expert instructor pilots in analyzing simulator-generated trainee performance data. The two research questions addressed in our work are

1) How do expert instructors use simulator flight data to describe task performance and diagnose pilot errors?

2) How can these analysis methods be mirrored in a computer-based system?

Our work uses high level formal methods *drawn from computer* pattern recognition and artificial intelligence to model the knowledge and procedures used by expert instructors.

Our long-term goal is to develop knowledge- based computer methods for performance analysis which emulate functions of expert instructors at three levels - during a flight (for real-time guidance), after a flight *(for critical debriefing on* the current flight trial), and across flights (for summative evaluation). The present work is focused on development of post-flight performance analysis methods. The work comprises four major tasks.

1) specification of a task domain and a set of flight tasks to provide a data base *for* guiding the model development and testing its application.

2) modification and extension of our ORLY flight simulation facilities to provide suitable flight dynamics, controls, instruments and other displays required for performing the flight tasks.

3) instructor analysis for deriving significant performance features and canonical errors in task performance.

4) functional design of computer algorithms for use in a post-flight performance analysis model.

This work is described in the next sections of the report.

SECTION II

OVERVIEW AND CONCEPTUAL FRAMEWORK

Automated trainers for complex real-time decision making and control tasks should have powerful task simulation and performance analysis facilities. Typically, however, sophisticated simulation and unsophisticated performance analysis are found in the same training system. The principles underlying effective task simulation methods are better understood and the technology more highly developed than for automated performance analysis methods. Thus, using well-established methods, all aspects of complex tasks such as instrument approaches or radar intercepts -- flight dynamics and control, radar operation, visual (through-the-window) display, ground control, and voice communications -- can be simulated even on minicomputer-based systems, at a high level of training fidelity. Using new knowledge-based methods developed expressly for this purpose, effective performance analysis capabilities can also be implemented on such systems. That is the principal goal of the work reported here.

Some automated task trainers are little more than task simulators, providing minimal performance analysis and task assignment facilities. Several state-of-the-art performance analysis systems compute aggregated measures of performance such as average absolute error or percent time within tolerance for specified parameters within each task segment. These measures are used as inputs to a scoring algorithm for task assignment. A high score advances the trainee to a new task, a low score sends him back to a previous task, and an intermediate score directs a repetition of the current task. This kind of analysis procedure sometimes uses complex data aggregation and parameter measurement algorithms. But the inherently statistical nature of the computed measures means that this approach will often fail to

3

recognize the specific features of performance, including major errors which give rise to the total aberration.

Automated trainers sometimes employ performance analysis methods designed for purposes other than training, for example, for selection and/or screening, performance rating and/or ranking, or pilot/vehicle system design and engineering. Even the most advanced and sophisticated of such systems are lacking in the key, central aspects of performance analysis required for training. They are not designed to recognize and describe performance errors, patterns, and events and they are not diagnostically oriented. These are prerequisites for any analysis which is to provide intelligent guidance and prescription, whether on a surface level or in a deep adaptive way.

An adaptive training system has several major functional modules:

- Task demonstration module
- Task simulation module
- Performance analysis module
- Diagnosis module
- Task assignment module.

These modules, as well as their interrelations, are schematically shown in Figure 1. Their functions are as follows.

2.1 Task Demonstration

This module explains and demonstrates to the trainee the appropriate performance of a specified task.

2.2 Task Simulation

This module provides a dynamic model of the task environment. It sets up the current training task and is responsible for the acquisition, storage, and retrieval of raw problem data generated by the trainee's attempt to perform the

```
        ┌──────────────────────┐
        │  Task Performance    │
        │   Demonstration      │
        └──────────────────────┘
                  ┆
                  ┆
                  ∨
        ┌──────────────────────┐       ┌──────────────────────┐
        │       Task           │──────▶│    Performance       │
        │    Simulation        │       │     Analysis         │
        └──────────────────────┘       └──────────────────────┘
                                                   │
                                                   ∨
                                        ┌──────────────────────┐
                                        │      Diagnosis       │
                                        └──────────────────────┘
                                                   │
        ┌──────────────────────┐                   │
        │       Task           │◀──────────────────┘
        │    Assignment        │
        └──────────────────────┘
```
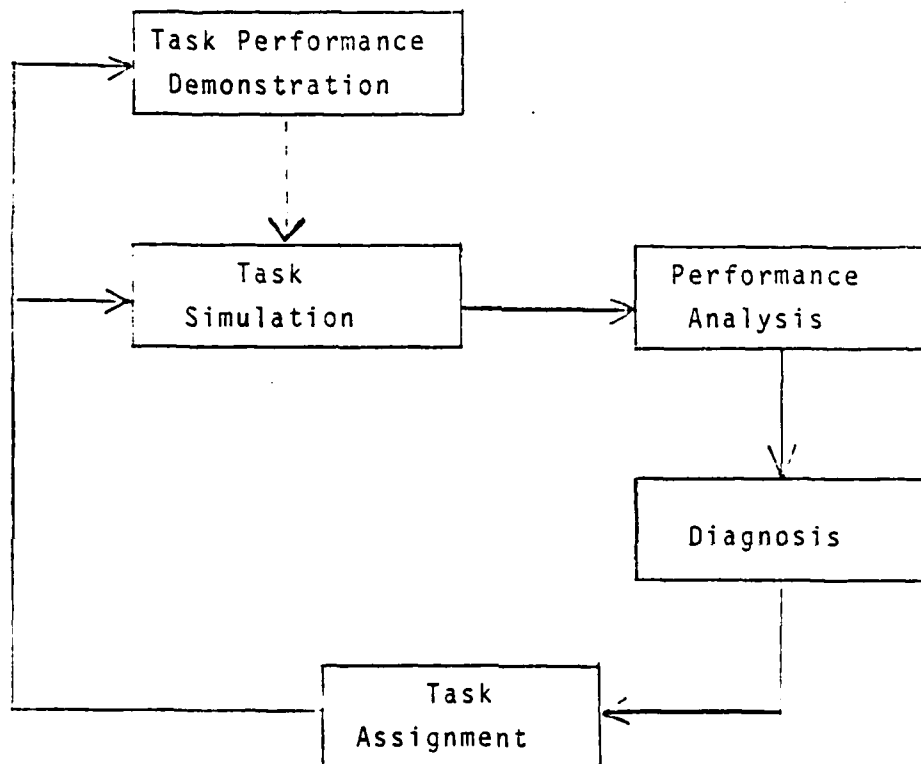
Figure 1. Functional Diagram of
Generic Training System

task.

2.3 Performance Analysis

Performance analysis can be done at three distinct levels: during the task (within task), after the task (post- task), and across tasks (inter-task). Within-task performance analysis means high-speed real-time analysis of errors during the course of task execution. This kind of analysis is necessarily shallow. It is designed to detect major surface errors and to provide immediate in-line guidance or warning information to the trainee or the instructor. Post-task analysis is done after task execution and has more ambitious goals. This deep evaluation seeks to detect the significant features and properties of errors, and to diagnose underlying difficulties and plausible reasons for them.

The accuracy and depth of post-task diagnostic inferences is significantly restricted by the limited amount of performance data generated on a single task trial. Inter-task performance analysis uses performance data generated by a trainee on several task trials, to provide a more stable, reliable, statistically sufficient base for evaluation. This kind of analysis is required to provide truly adaptive training.

The performance analysis module consists of three submodules:

1) Within flight analysis

        This module, also called the guidance and warning
        system (GWS), produces advisory messages when the trainee
        makes major errors. It interacts closely with
        the task simulation module.

2) Post-flight analysis

        This module consists of two submodules:

        a) The performance description submodule which

6

generates precise and detailed descriptions of common
task-specific   (i.e., canonical) errors, error patterns,
error-prone behaviors, and error contexts.

b) The interactive query submodule which attempts
to elicit additional information from the trainee
to help distinguish knowledge/skill deficiencies
from effects of inattention and to otherwise
resolve ambiguities among alternative interpretations
of performance behaviors.

3)   Inter-flight analysis

This module consists of two submodules:

a) The history submodule which integrates the current
trial with performance descriptions from previous
trials and performance history on other tasks to provide
an updated history.

b) The inferential machine which generates a goal
directed student profile.

2.4 Diagnosis

This module contains two submodules:

1) the diagnostic inference submodule which generates
goal-directed hypotheses about the trainee's skill
deficiencies and conceptual gaps.

2) the hypothesis generation submodule which sets up
tasks and task sequences designed to confirm or
disconfirm hypotheses.

2.5 Task Assignment

The task assignment module has two distinct functions - the
choice of an appropriate task, and the choice of the training
mode within which the task is to be performed.   There are at
least three significantly different training modes in any
comprehensive training program -- demonstration, guided practice,
and solo trial. The trainee can be shown how to perform the task
(demonstration mode); he can try to perform the task under close

monitoring and supervision by an instructor (guided practice mode); or he can be given complete freedom and responsibility for performing the task on his own (solo trial mode). Proficiency is often acquired through activities involving a mix of these three modes.

This module contains two submodules:

1) the task selection submodule chooses an appropriate task from the set of specified tasks within the domain or, if necessary, generates requirements for a new task.

2) the generative task building submodule which, given requirements for a new task, generates the detailed task specifications from basic subtask building blocks.

A comprehensive functional diagram of the system is shown in Figure 2.

The framework outlined so far applies very generally to all task domains. Given a particular task domain, such as the category of flight tasks, a finer and more detailed specification of the modules is desirable. In this framework the organization of the modules has to take into consideration the features common to all the tasks in the category. In particular it is important to isolate the submodules that are maintained invariant across the targeted tasks. Once these submodules are specified, they can be designed and implemented once and for all. The remaining submodules are the ones specific to each task and must be customized accordingly.

In this research effort, initially planned for four years, we have focused on the area of flight tasks; and particularly VOR and ILS approaches. We have isolated, designed and implemented some of the task invariant performance analysis submodules. We describe these in Section 3. We have also developed initial versions of some of the task-specific simulation and performance analysis submodules for ILS approaches. These are described in
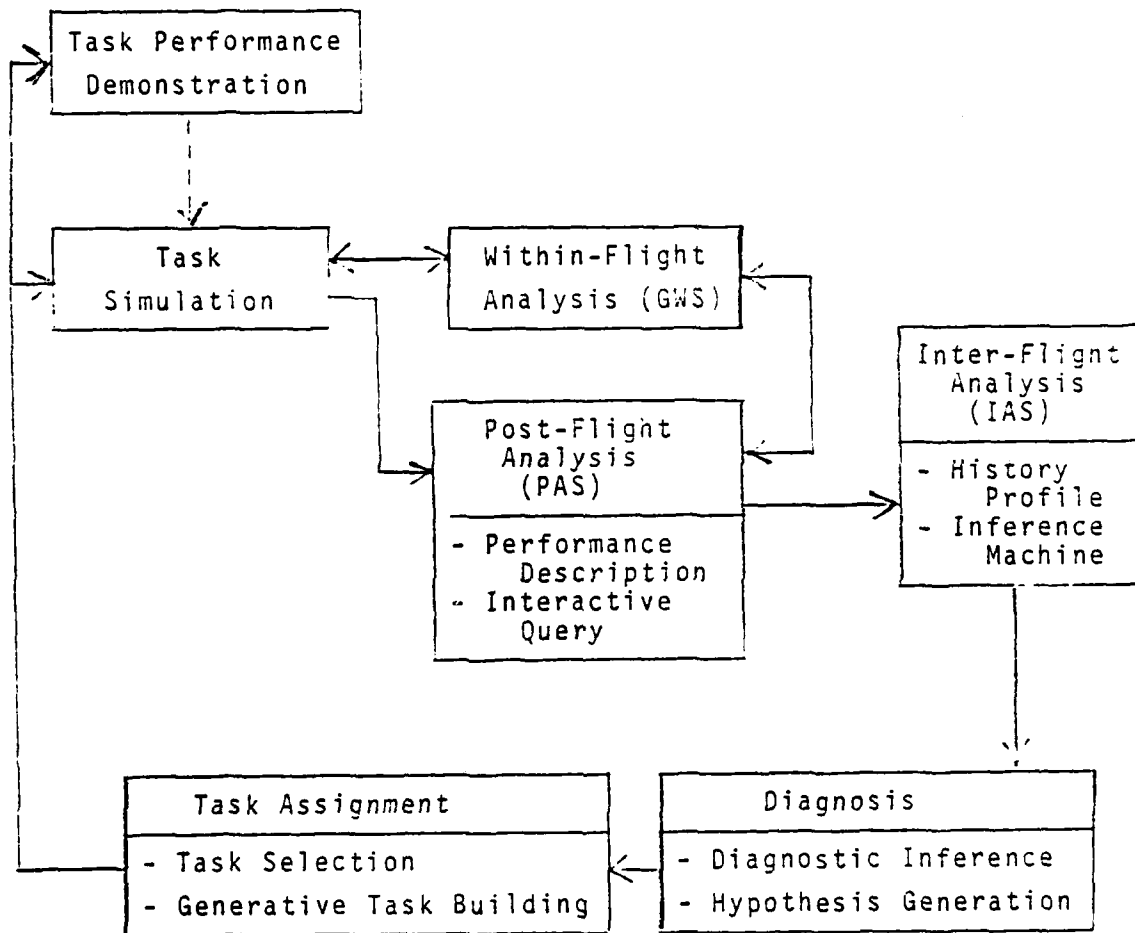
```
┌──────────────────────┐
│ Task Performance     │
│ Demonstration        │
└──────────────────────┘
```

```
┌──────────────────┐        ┌──────────────────────┐
│     Task         │        │ Within-Flight        │
│  Simulation      │        │ Analysis (GWS)       │
└──────────────────┘        └──────────────────────┘
```

```
                            ┌──────────────────────┐
                            │ Inter-Flight         │
                            │ Analysis             │
                            │ (IAS)                │
         ┌──────────────────┤──────────────────────┤
         │ Post-Flight      │ - History            │
         │ Analysis         │     Profile          │
         │ (PAS)            │ - Inference          │
         ├──────────────────┤     Machine          │
         │ - Performance    └──────────────────────┘
         │     Description
         │ - Interactive
         │     Query
         └──────────────────┘
```

```
┌───────────────────────────────┐   ┌──────────────────────────────┐
│ Task Assignment               │   │      Diagnosis               │
├───────────────────────────────┤   ├──────────────────────────────┤
│ - Task Selection              │   │ - Diagnostic Inference       │
│ - Generative Task Building    │   │ - Hypothesis Generation      │
└───────────────────────────────┘   └──────────────────────────────┘
```

Figure 2. Functional Diagram of Comprehensive
Adaptive Training System

9

Section 4.

SECTION III

TASK-INVARIANT PERFORMANCE ANALYSIS SUBMODULES

In general, task invariant submodules can be isolated within all modules. In the case of the post-flight performance analysis module, the performance description submodule includes a task-specific component (for detection of salient task features and canonical errors) and a task-invariant component (the generic flight description module). We devote the rest of this section to the description of this latter module.

The module takes as its input a set of raw data (usually data points from the A/D channels) and it produces a high level description of the activity described by the low level data.

This module contains a set of parameter description modules (PDM), one for each task relevant parameter. (For ILS approaches, for example, these parameters include heading, altitude, airspeed, rate-of-turn, rate-of-climb, glide slope, localizer, and power.)

Figure 3 shows the structure of a PDM. Typically, a PDM contains four submodules:

- raw data aggregation
- mode segmentation
- semantic segmentation
- context segmentation.

Each is described in turn.

3.1 Raw Data Aggregation

The data are input at a very low level (large temporal sets of points) and cannot be usefully analyzed in this form. Therefore, it is necessary to reduce the amount of data while preserving the information conveyed, The data aggregation thus
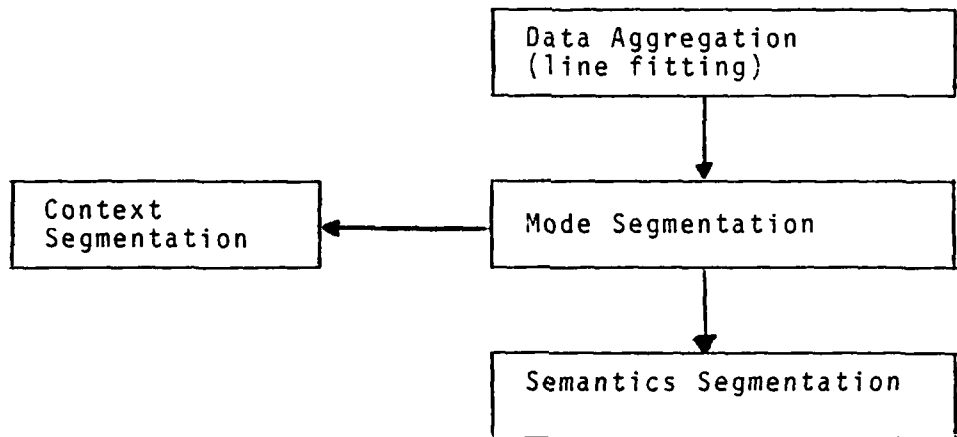
```
                                    ┌──────────────────────┐
                                    │ Data Aggregation     │
                                    │ (line fitting)       │
                                    └──────────┬───────────┘
                                               │
                                               ▼
┌──────────────────┐            ┌──────────────────────┐
│ Context          │◄───────────│ Mode Segmentation    │
│ Segmentation     │            │                      │
└──────────────────┘            └──────────┬───────────┘
                                           │
                                           ▼
                                ┌──────────────────────┐
                                │ Semantics Segmentation│
                                │                      │
                                └──────────────────────┘
```

Figure 3.   Structure of a PDM.

called for should be:

- conservative, i.e., no useful detail is lost
        in the process
- complexity reductive, i.e., it should return a data
        representation that is simpler and easier to
        process than the original representation
- expert analogic, i.e., it should be easy to move
        between this (computer) data representation and the
        representation used by the expert instructor when
        analyzing the flights.

The performance analysis system aggregates raw data by fitting the data points for each parameter with a set of line segments.

This linear decomposition satisfies the above three mentioned criteria since, 1) flight tasks are defined in terms of linear functions of flight parameters such as climbs or descents at constant rate, standard rate turns , etc. If a primary parameter is not linear, one of its higher derivatives almost invariably is.

2) the line fitting algorithm returns a small (relative to the number of points) number of segments which are simple interpretable constructs. The operation of this algorithm is described later in this section.

3) when analyzing time plots of flight parameters all our instructors used some form of straight line fitting as a first step in organizing the data.

In an earlier research effort, we systematically investigated line fitting algorithms to determine which would be most suitable for our flight analysis work.   (Linear Approximation Techniques for Fitting Flight Performance Data, Final Report N61339-77-M-1286-1, July 1977.) The algorithms for linear fitting fall into two general classes. One can perform an

incremental search for linear fits, detecting minimum size line segments, then possibly extending or combining them to create larger but less precise fits. Alternatively, one can take a global approach aggregating data on all points to be fit and then doing all fitting at once.

An algorithm for the latter approach is based on a phase plane analysis in which each point to be fitted is mapped as a curve onto a plane, each of whose points represent a line. The curve associated with each data point corresponds to the set of lines which pass through it. Linear fits thus correspond to clustering of intersections of these curves. This method is effective but computationally extremely expensive because the entire mapping of points to the phase plane must be performed before determination of linear fit can be made.

The least mean squares line fitting algorithm is representative of the incremental line finding approaches. We considerably extended this method, since the classical least mean squares algorithm is not useful for describing flight data. In this extended form it has proved to be highly suitable. The least mean squares method, as generally used, fits an entire set of data with one line segment, giving the mean square deviation as a measure of fit. We want to fit a set of data with a sequence of line segments. We use the mean square deviation as a cutoff criterion. However, we have modified the classical goodness of fit criteria . In the usual application of the least mean squares algorithm the vertical distance between each point and the line fit is used as the basis for calculating goodness of fit. In our work we need to consider the deviation as based on the perpendicular distance between point and line fit. To do otherwise would strongly bias the algorithm against lines of large slope. In particular, as described later, segments in the change legs would be favored. The correction is very simple -- instead of a constant value C uniform for all line segments, we

use as criterion an expression $C(1+a*a)$ for adequate fit, where a is the slope of the line fit. The use of the expression $C(1+a*a)$ now makes our fitting rotationally invariant - the same figure of merit is obtained regardless of slope.

A second important problem that is addressed in our algorithm is overshoot of line fit. We keep on adding points to our set of points which fit a line segment until the mean square deviation exceeds some prescribed value. If we arrive at a fit better than this value in some region, we have a "reservoir" which enables really poorly fitting adjacent points to be included in that line fit. We therefore backtrack once a fit is found, discarding points whose square deviation exceeds $KC(1+a*a)$, where K is a cutoff factor. We have found that a value of K=0.5 works well on all our flight data. This procedure eliminates overshoot and guarantees a good fit, even at the ends of the segment. The discarded points are likely to be the initial portion for the next segment. Thus, this technique typically improves the fit for the next segment as well as for the current one. Also it assures that no useful information is lost in the aggregation process.

A third difficulty is due to the fact that a least mean square line fitting algorithm cannot produce a reasonable continous fit in regions of sharp variation. We have therefore incorporated for those cases an extra, rougher fitting mechanism which deals with remaining gaps. As soon as a line fit is found by the line fitting algorithm the region between it and the previous line fit is examined. This gap is then fitted with a set of line segments, the only requirement being that of monotonicity - each monotone segment of data points is considered to lie on a line.

Our line fitting algorithm is presented in some detail in flowchart form as Figure 4. The associated parameters are

defined in Table 1.

The line fit algorithm is real time. The computational speed of the algorithm is approximately proportional to the number of data points and is not strongly dependent on the number of segments achieved by the fit.

3.2 Mode Segmentation

The raw aggregation submodule produces a parameter description consisting of a (relatively) short sequence of segments. Thus the size of data is significantly reduced without losing semantically relevant information.

So far the data description process has been highly syntactic. Next the data is further segmented using semantic criteria pertinent to the set of actions involved in performing the task.

In the case of most flight tasks, performance on any parameter can be characterized as being in either of two modes: maintenance (a parameter is kept at approximately the same level), and change (a parameter is changed from one level to the other). For example, the airspeed can be maintained at cruise level or can be changed to a higher or lower setting.

The mode segmentation submodule produces a sequence of change and maintenance legs for all flight tasks. The precise meaning of a change or maintenance leg is task specific, i.e., it depends upon a set of task specific parameters in the manner described next.

We begin by defining a change segment. A change segment is one which is greater than t seconds duration and whose slope is comparable in magnitude to the standard rate of change, SR, for the parameter, i.e. slope $= c*SR$. The two constants c and SR depend upon the parameter and the task in question. For example,
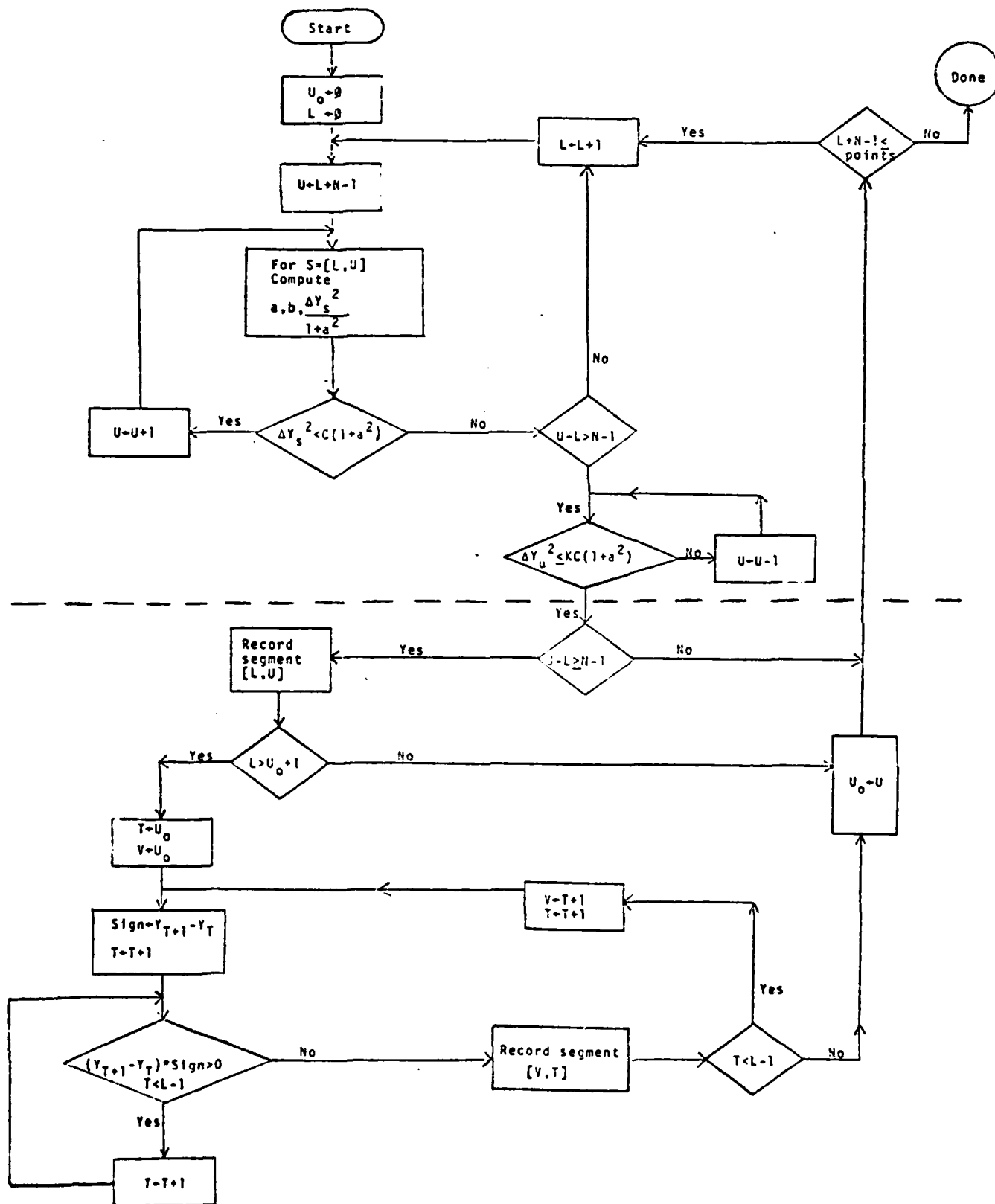
16

Figure 4.   Least Mean Square Line-Fitting Algorithm

17

TABLE 1.   PARAMETERS FOR LEAST MEAN SQUARE
           LINE-FITTING ALGORITHM

S   Line fit candidate interval S=[L,U]

L   Lower bound of line fit candidate interval

U   Upper bound of line fit candidate interval

N   Minimum number of points for a valid line fit

Points   Total number of data points

U_0 Upper bound of previous line fit

a,b   Slope, Intercept of line fit candidate

ΔY_s^2 Mean square deviation of entire candidate segments

$\dfrac{\Delta Y_s^2}{1+a^2}$ Mean square deviation perpendicular to line fit in interval s

C   Criterion for valid line fit - maximum mean square deviation allowed

K   Cutoff factor for eliminating points at end of line fit candidate

ΔY_u^2 Square deviation of endpoint from line fit candidate

V,T   Lower and upper bound of secondary line fit

Sign   Sign of difference of ordinates of adjacent points T, T+1

in our tasks we have used t=15 seconds, c=.8 and SR(heading) = 180 degrees/minute.

The leg partitioning algorithm has three phases whose operation is as follows.

1.   Identify and mark all "change" segments as anchors.

2.   Extend each such anchor segment in both directions as far as possible by adjoining those neighboring segments which have critical slope value ($|slope|$ > c*SR).   This set of contiguous segments constitutes a change leg (a turn leg for heading, a climb or descent leg for altitude, an increased or decreased airspeed leg for speed).

3.   The sequence of segments between two successive change legs is labeled as a maintenance leg, or base leg for short.

Although very simple, this algorithm has been effective in making the same identification of legs as those made by instructors on all recorded flights that have been analyzed.

3.3 Semantic Segmentation

The input to this segment is a sequence of maintenance and change legs. While this description gives some idea about the subject's performance within some parameter, it is not fine enough to be analyzed and diagnosed. The data representation produced by the semantic segmentation submodule makes it easier to recognize error and control patterns, to find correlations among patterns within a parameter, to describe patterns and contextual relationships involving two or more parameters, and to interpret these in diagnostic terms. We have modeled these aspects, which are characteristic of expert instructor analysis, in our program. The elements of our formal description of flight segments are presented in this section and methodology is developed for generating a detailed semantic description of the

segments comprising a flight parameter.

The first step toward assigning meaning to the line segment description of flight data is to tag each segment with a flight attribute. The attribute types in our model reflect both the standard operational meanings of the flight parameters and their prescribed constraints as imposed by the flight plan, i.e., their specified tolerance regions.

These tolerance regions are specific for each parameter and for each task.

The attributes for this initial level of semantic representation are defined using the following segment measures:

      a. the positions of its extremities relative
to the prescribed tolerance region,

      b. the magnitude and sign of its slope,
and
      c. its temporal length (i.e., projection on
the time axis).

Using simple functions of the segment measures, each maintenace leg segment is characterized uniquely as one of four segment types -- an error segment, a correction segment, a stable segment, or a very short segment. These are defined relative to tolerance regions as follows.

- A very short segment is a segment of temporal length < t' seconds.

- An error segment is one with right extremity outside the tolerance region, slope < c^* SR and temporal length > t' seconds.

- A correction segment is one with left extremity outside the tolerance region, slope < c^*SR, and temporal length > t' seconds.

- A stable segment is one with both extremities inside the tolerance region, and temporal length > t' seconds.

These definitions correspond intuitively to their names. Examples of the four types of segments are shown in Figure 5. Figure 5 shows examples of a variety of error segments and correction segments which have different orientation and passage relative to the tolerance region. The constants in the definitions again vary with the parameter and the task in question. To make additional semantically useful characterizations of different types of error segments, the slope and time length measures are further quantified as follows. An error segment is shallow if it has a small slope value relative to the standard rate for the parameter; otherwise it is a sharp segment. The criterion for a shallow segment is slope $<1/3$ SR. Similarly, an error segment is either long or short, according to whether or not it crosses the center line of the tolerance region. Using these characterizations we are able to distinguish three significantly different types of error segments called drifts, deviations, and overshoots.

A drift is a shallow error segment. Intuitively this type of error suggests unintentional change in a parameter value due to inattention. Another type of error, called a deviation, is a sharp error segment. This type of error suggests a deliberate control change in a parameter. Drifts and deviations may further be described as short or long according to whether or not they cross the center line of the tolerance region. The third type of error segment, called an overshoot, is a change segment causing the maintenance leg to begin outside the tolerance region. Since an overshoot is actually part of a change leg, it could be discussed in that context. However, since it significantly affects the pilot's performance during the maintenance leg, we include it as a maintenance leg error.

There are also three distinct types of correction segments, called undercorrections, corrections, and overcorrections. As correction segments, they all originate outside the tolerance region. Undercorrections remain on the same side of the center line that they start from. Corrections go beyond the center line into the other half of the tolerance region but stay within it. Overcorrections exit the tolerance region on the opposite side from which they start. Overcorrections look like overshoots but are not change segments - unlike overshoots they originate within a maintenance leg. Examples of these characteristic types of errors and corrections are shown in Figure 6.

3.4 Context Segmentation

The parameter description module includes another submodule with important semantic implications - the context segmentation submodule. The underlying notion is that flight performance tasks require a pilot to manage several flight parameters - heading, altitude, airspeed, power, bank, etc. - jointly and coherently. Thus a complete description of performance must simultaneously take into account all the parameters, not just one. An error on some flight parameter is often due to a student's preoccupation with another parameter; drifts in heading, for example, are frequently due to difficulties in altitude transitions.

Part of the description of each segment, then, is information concerning the pilot's concurrent state in the context of the associated parameters. For these purposes the legs (maintenance or change) of an associated parameter are divided in three sections: entry section, within section and exit section. The segmentation is done by specifying temporal lengths for the entry and exit sections. Thus we have: maintenance entry time (mnt), maintenance exit time (mxt), change entry time (cnt) and change exit time (cxt). All these values are
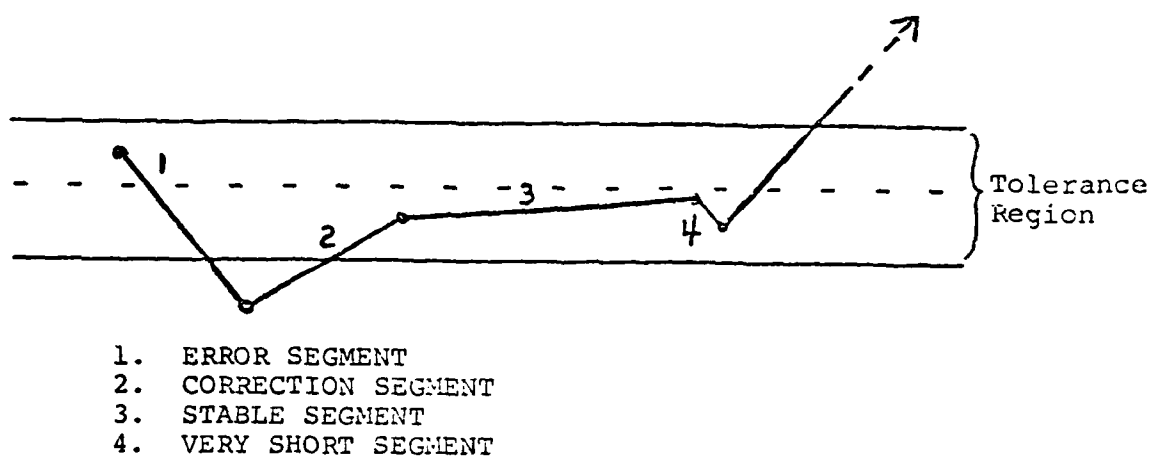
1. ERROR SEGMENT
2. CORRECTION SEGMENT
3. STABLE SEGMENT
4. VERY SHORT SEGMENT

Figure 5. Segment Types

ERROR SEGMENTS

I/S to O/S   O/S to O/S

CORRECTION SEGMENTS

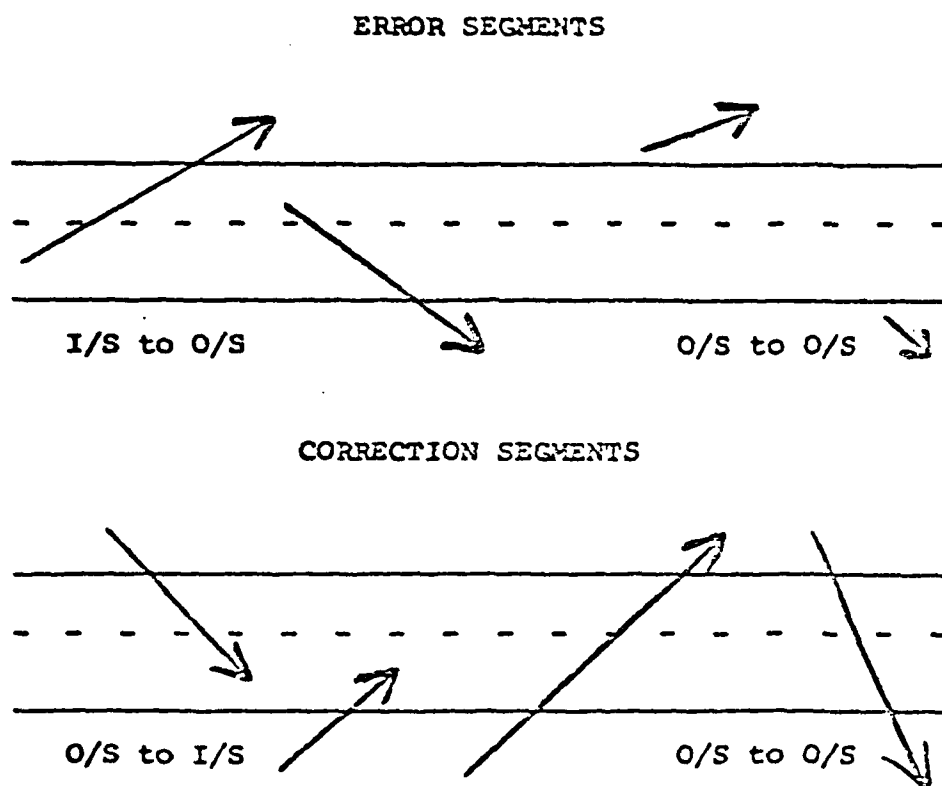O/S to I/S   O/S to O/S

Figure 6. Error and Correction Segments

parameter-and-task-specific.

Given the temporal coordinates, {A B} and {C D}, of maintenance and change legs, the following such states are defined:

a. Maintenance State (being within a maintenance leg): {A+mnt B-mxt}

b. Change-increase State (changing to a higher setting): {C+cnt D-cxt}

c. Change-decrease State (changing to a lower setting): {C+cnt D-cxt}

d. Transition States

- from maintenance state to change-increase state: {B-mxt C+cnt}

- from maintenance state to change-decrease state: {B-mxt C+cnt}

- from change-increase state to maintenance state: {D-cxt A+mnt}

- from change-decrease state to maintenance state: {D-cxt A+mnt}

e. Multiple State (two or more of the above states)

Figure 7 illustrates the partitioning used to define transition intervals for a flight parameter.

All the information needed to describe a flight parameter as a series of segments has been specified. The concepts used as segment attributes -- error segment (drift, deviation, overshoot); correction segment (undercorrection, correction,
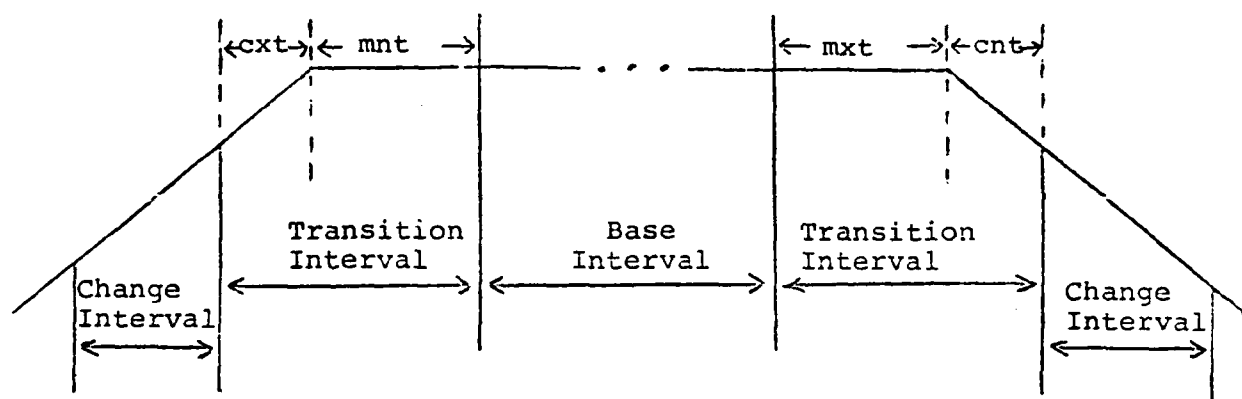
Figure 7.   Definition of Transition Intervals.

overcorrection); stable segment; very short segment; context
(maintenance, change, transition) -- have been defined in a
precise and explicit fashion. These concepts will be used to
recognize the canonical errors associated with the task.

SECTION IV

TASK-SPECIFIC SUBMODULES

In this section we give a brief description of a particular set of instrument flight tasks, ILS approaches, as background for discussion of two task-specific components of the performance analysis model, canonical error description and task simulation.

4.1 ILS Approaches and Errors

A standard ILS approach procedure includes the following procedural components.

1.   Initial approach between the final enroute navigation point or final nav fix and the outer marker (OM).

2.   Interception of the ILS path at the OM.

3.   Tracking outbound on the ILS path from the OM.

4.   Procedure turn.

5.   Tracking inbound on the ILS path to the missed approach point.

6.   Landing (from the missed approach point inbound to touchdown).

The first four phases comprise the initial approach; the last two comprise the final approach.   Phases 3 and 4 may be omitted in some situations.

Each phase involves a number of navigation procedures and provides extensive opportunities for various types of errors. Basic instrument procedures can often be executed mechanically as tracking tasks.   This is no longer possible when there is a navigation component.   Navigation instruments require more

complex and subtle interpretation than flight instruments. The
pilot must conceptualize his position in space relative to
various points on the ground such as marker beacons, the airport,
and the ILS approach path.

Representative canonical errors for each of these phases
include the following.

Phase 1. Start of Initial Approach

1. Failure to select appropriate radio channel.

2. Failure to select appropriate ambiguity
    switch position for a front or back course
    ILS as appropriate.

3. Confusions in response to vector commands
    (e.g., between right turn and left turn,
    0 degrees and 360 degrees, and in
    termination of relatively long turns such
    as 180 degree turns).

4. Failure to halt descent at the minimum
    assigned altitude.

5. Misinterpretation of command "cleared for
    the approach" resulting in use of the final
    approach altitude instead of the (higher)
    initial approach altitude.

Phase 2. Interception of ILS Path

1. Failure to acquire an appropriate intercept
    angle between the actual path of the aircraft
    and the ILS path.

Phase 3. Outer Marker Outbound

1. Failure to bracket the course and center
    the localizer or course deviation
    indicator (CDI).

2. Failure to attain and maintain the OM
    inbound altitude.

3. Failure to fly outbound for at least one
    minute from the OM.

4. Failure to verify operation of the marker

beacon receiver.

Phase 4. Procedure Turn

1. Failure to maintain an appropriate rate of
    turn (standard rate but not more than
    30 degrees of bank).

2. Failure to attain and maintain procedure
    turn altitude.

Phase 5. Tracking Inbound

1. Failure to bracket the ILS course inbound
    and center the CDI prior to reaching the OM.

2. Failure to maintain approach altitude
    prior to reaching the OM.

3. Failure to establish the appropriate rate
    of descent prior to reaching the OM.

4. Failure to establish appropriate approach
    speed including flap extension prior to
    reaching the OM.

5. At the OM, failure to extend the landing
    gear and initiate descent at the appropriate
    rate.

6. Failure to keep the CDI and glide slope
    indicator (GSI) centered for the descent.

7. Failure to maintain appropriate airspeed
    during descent.

8. Failure to initiate missed approach
    procedures if any of the approach parameters
    (e.g., CDI, GSI) are exceeded beyond
    permissible limits.

9. Failure to recognize the OM or the missed
    descent altitude as appropriate (i.e.,
    "busting your altitude") and failure to
    initiate the missed approach procedure.

10. "Chasing the needle" with respect to either
    the CDI or the GSI.

Phase 6. Landing

1. Failure to effect transition from instrument
    to visual approach. (This is a classical
    canonical error, often noted by failure to
    decrease the rate of descent.)

2. "Ducking under" the glide path, e.g., to get
    beneath the clouds.

Further types of errors commonly occur when there are
significant wind conditions or when the flight or navigation
instruments develop operational faults. Typical examples of such
instrument faults and associated canonical errors include the
following.

Phase 1. Start of Initial Approach

1. Inoperative CDI, leading to failure to locate
    the ILS path.

2. Inoperative marker beacon receiver, leading to
    failure to locate the OM.

Phase 2. Interception of ILS Path

1. Inoperative DME (distance measuring equipment)
    coupled with a tailwind, leading to overshoot
    of the ILS path. (This becomes probable if
    the pilot relies solely on the DME for ground
    speed, distance, and timing information.)

Phase 3. Outer Marker Outbound

1. Faulted directional gyroscope (DG) not noted by
    pilot. If the pilot is not watching the CDI
    closely, it may leave the box. If the fault
    causes the DG to precess slowly (a few degrees
    a minute), and the pilot is not matching his DG
    with his CGI, he may not catch the failure at all
    and he could fly off course.

Phase 4. Procedure Turn

1. Localizer failure. If the pilot relies on the
    localizer to indicate proximity to the ILS
    course during the procedure turn, without
    monitoring the DG, clock and turn coordinator,
    he may fly through the ILS course or continue

circling.

Phase 5. Tracking Inbound

1. Inoperative GSI. If the needle is stuck,
    especially in an intermediate position, and
    the pilot does not notice this (and fails to
    spot the "off" flag), he is likely to chase the
    needle.

2. Inoperative attitude indicator (AI). Even when
    this failure is noted, and even though the AI
    is not actually important in an ILS approach,
    many pilots find it very difficult to delete
    this basic instrument from their scan; as a
    consequence they have trouble following CDI
    and GSI information, and show erratic airspeed,
    heading, and altitude control.

3. Combined marker beacon receiver failure and GSI
    failure. If the pilot relies solely on the
    ADF to indicate OM passage, and the GSI failure
    is insidious (for example, bouncing around
    slightly about the central position), the pilot
    may erroneously initiate the descent.

4. Failure of pilot-static system, including the
    airspeed indicator, vertical speed indicator, and
    altimeter. Pilots may lose control of the
    aircraft in this situation.

A fairly comprehensive list of ILS approach errors,
including several others induced by failures of one or more
faulted instruments, can be enumerated. In each case,
significant deviations from the flight plan are clearly shown as
tolerance violations in one or more task parameters. A given
canonical error is characterized by a very specific pattern of
such deviations. The module for canonical error detection
consists of pattern matching procedures designed to recognize
these task-specific error patterns. The task simulation module
generates the task parameter data which constitute the starting
point for the analysis, along the lines described in Section III.
These two modules are described next.

4.2 Task Simulation

The task simulation environment was provided by the ORLY flight simulator.   ORLY includes facilities for simulating a large class of instrument flight tasks.   The ORLY simulation system has four major logical components: 1) the aerodynamics computation,   2)   the   instrument   display,   3)   the   visual through-the- window display, and 4) the user interface display. The aerodynamics computation is performed according to the dynamic flight model which is stored in the computer program. The current ORLY instrument display shows the standard aircraft instruments including a navigation panel with ILS, ADF, DME, and marker beacons.   The instruments occupy the lower half of the display; the upper half is used to show the through-the-window view.   Currently this includes schematic representations of runway and ground facilities (to enable visual take-offs and landings).   The ground track map may also be displayed either during or after the flight. Following the flight, charts showing pilot performance on specified task measures may be displayed. The user interface consists of the flight controls (stick, throttles, switches, etc.) and other controls needed to operate the system (changing ceiling, winds, etc.).

A detailed description of the ORLY system, components, and facilities for display, recording, and playback is given in the NTEC report "Higher Order Adaptive Systems" cited above, and in "ORLY   -   A   Minicomputer-Based   Instrument   Flight   Training Simulator," G. Lukas, W. Feurzeig, D. Cohen, Aviation Research Journal, Vol. II, July 1977.   One current version of the ORLY instrument display is shown in Figure 8, which also shows ground facilities and distant mountains.   The ORLY aerodynamics, instrument display, and user interface had to be modified as follows for use in the present research.

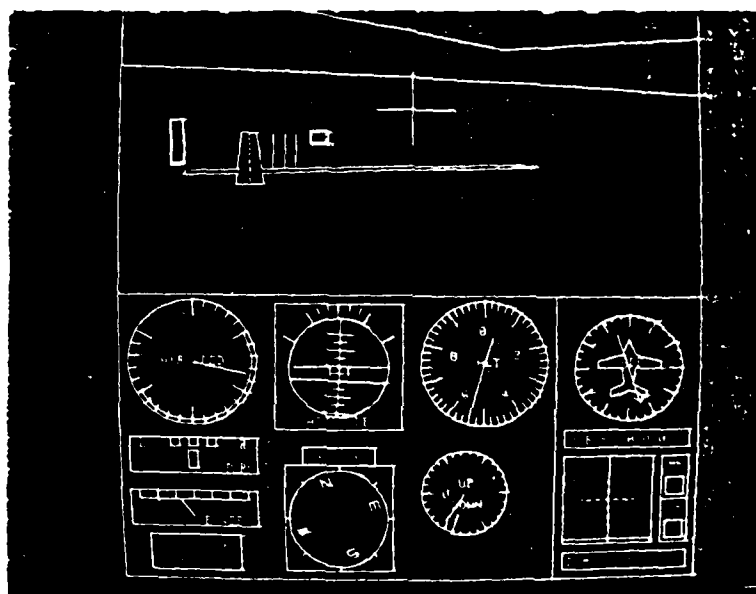The   original   aerodynamics   model   implemented   in   ORLY

Figure 8.   The ORLY Instrument Display.

simulated a light single-engine aircraft. This low performance flight dynamics was adequate for our previous research, which involved novice instrument pilots and relatively simple flight tasks including no navigation component. The present effort, however, involved experienced pilots and more complex flight tasks. Thus, the ORLY dynamics model was replaced by one which simulated an aircraft with much higher performance, the T-28. The dynamics equations used in this implementation were taken from "Simulation of T-28 Aircraft," O. Carpenter, I. Golovcsenko, and B. Newman, Technical Report NAVTRAEQUIPCEN IH-75, Naval Training Equipment Center, April 1967. Modifications were made as required, e.g., to remove the facilities for flags, landing gear and rudder, since these controls are not included in the present ORLY user interface, and to the thrust-speed dependency to improve flight performance.

We modified the ORLY flight and navigation instruments to make them more closely conform to those found in high performance airplanes. A new altitude indicator was implemented; the flat needle indicator was replaced by a moving ball display to simulate a gyro instrument. The scaling of the instruments was adjusted to reflect the higher performance of the T-28. The ORLY simulation time/real time ratio was changed from 2:1 to 1:1. Corresponding changes were made in the level and range of values of the pertinent instrument parameters. The digital clock was replaced by a dial clock, the reed-like power indicator was replaced by a thermometer-like instrument, and the size of the VSI indicator was increased. These modifications give better visual clues during the landing procedures.

The original user interface consisted of a system of throttle, joystick, and switches which simulated the aircraft control panel. These have been replaced by a more realistic control system. The joystick and the rotary throttle were replaced by a yoke and power quadrant which allow more precise

control movements, as required for the present tasks. Similarly, navigation instruments were updated to provide improved controls for landing tasks. All the instruments were mounted on a special panel connected to the computer system.

4.3 Canonical Error Detection Module

A standard output of the task simulation module is a flight chart showing pilot performance on a series of task-specific flight parameters. One such chart, generated on an ILS approach, is shown in Figures 9A and 9B. It shows the final phases of the approach. This chart includes time plots of the following parameters: rate-of-climb, rate-of-turn, heading, altitude, airspeed, power, glideslope, and localizer. The two plots include duplication of some parameters; the choice of grouping was made to aid analysis of closely related parameters. The parameters at the top of the chart (TO, DT, N, and PLBK) give respectively, the starting time (in frames), the time between frames, the total number of data points, and the identification of the flight run (in this case BCILS1). These flight data are essentially the inputs to the task-invariant analysis process described in Section III. A later phase of the analysis is responsible for detection of the specific features and canonical errors.

The flight data for each task parameter are processed to produce a performance description for that parameter in the form of a sequence of linear segments. These descriptions provide one major set of inputs to the error detection module. A second set of inputs consists in similar descriptions of task parameters, corresponding to the correct execution of the procedures prescribed in each phase of the flight plan (to within allowable error tolerances). A third set of inputs consists of descriptions of canonical errors, again in the same general form. Diagrams showing instances of the second and third kind of inputs
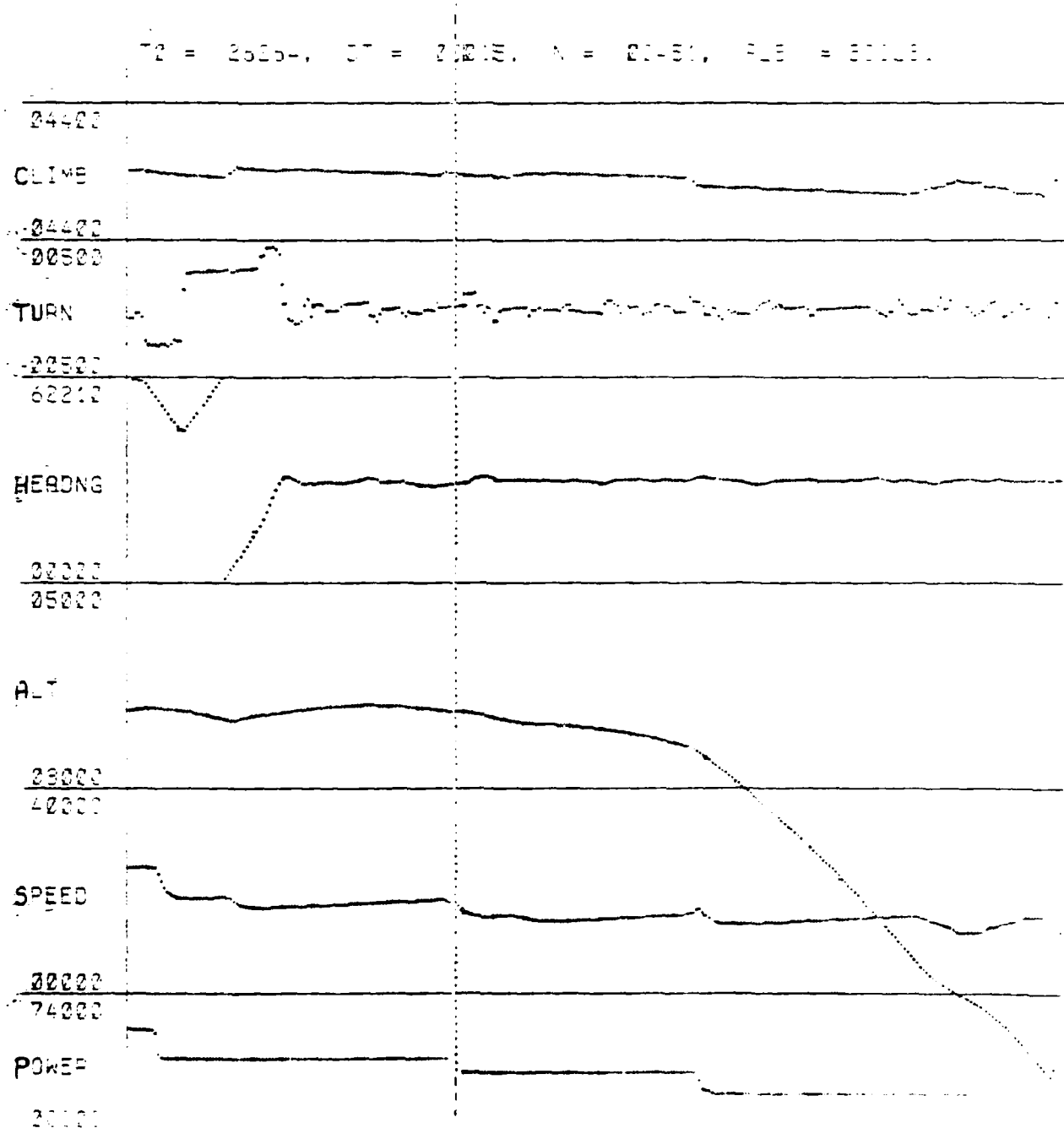
35

T2 = 3505-, CT = 2 2 E, N = 2 -5 , L = 3 .

04400

CLIMB

.04400
.00500

TURN

.00500
62210

HEADNG

00000
05000

ALT

23000
40000

SPEED

30000
74000

POWER

30000
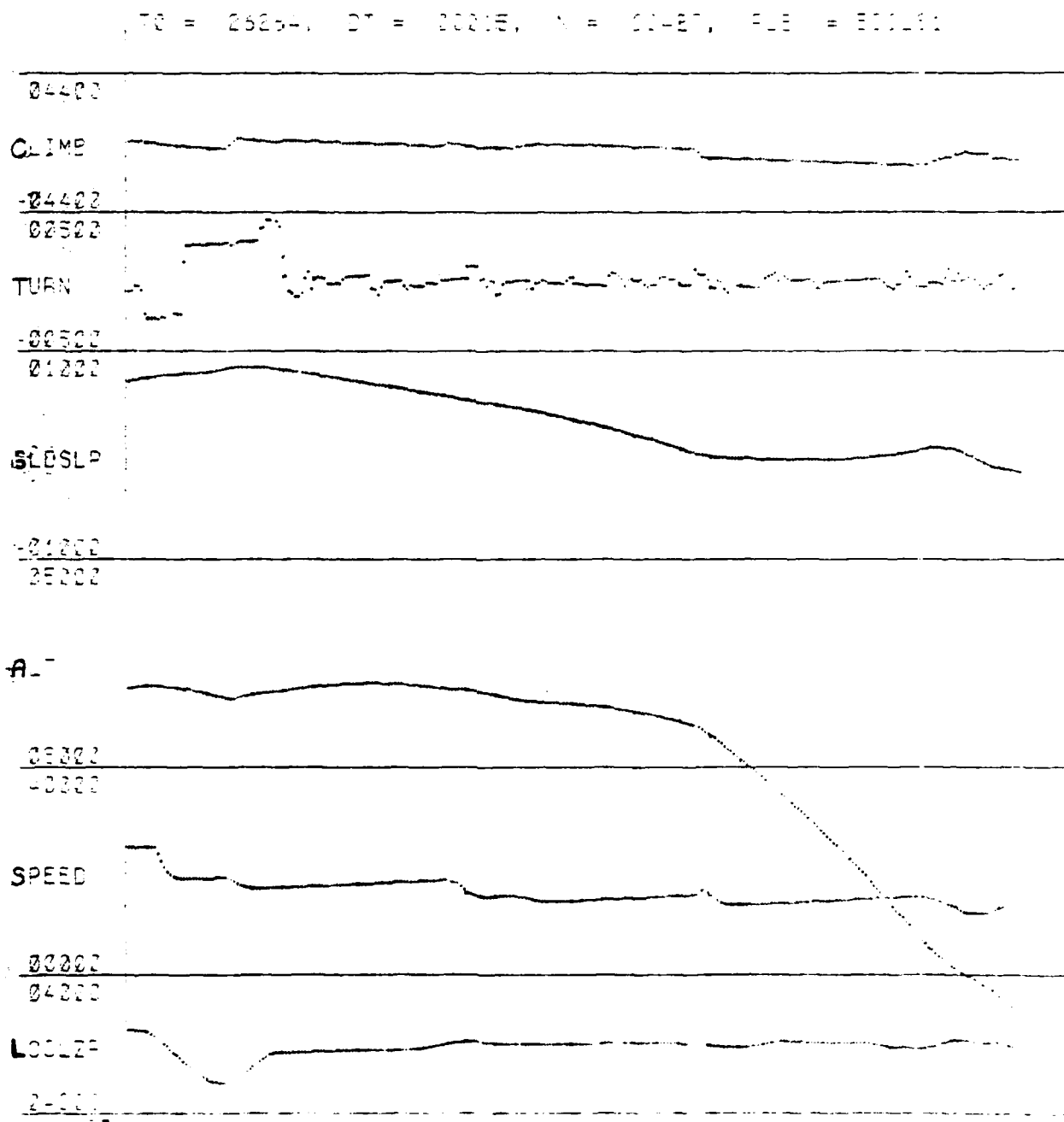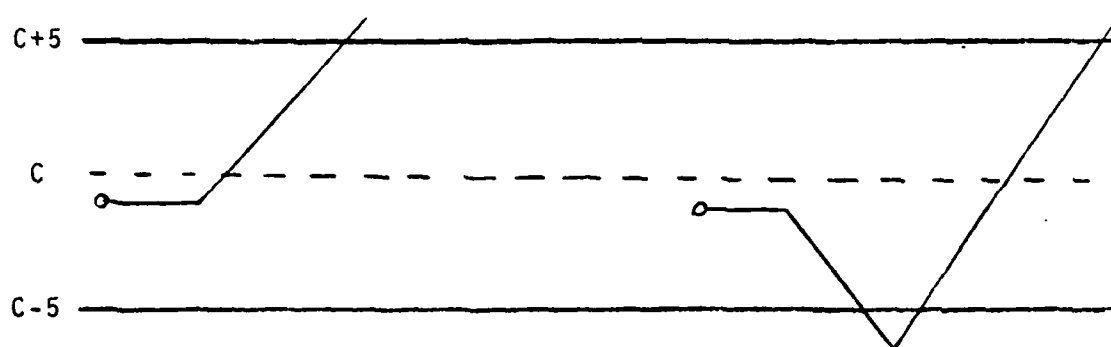
Figure 9A.   Flight Chart 1 - Flight 8CILS1

Figure 9B.   Flight Chart 2 - Flight BCILS1

are given in Figures 10 and 11.

Figure 10 shows the correct execution and a canonical error for two turn maneuvers, as required for example, in the ILS procedure turn. Figure 10A shows the proper initiation of the turn; figure 10B shows the classical canonical error of starting a turn in the wrong direction and then correcting back. These are diagrams of the heading parameter in which C indicates the initial compass direction and C+5 C-5 mark the allowable tolerance bounds of +5 degrees. In figures 10C and 10D the rate-of-turn parameter is shown, with the tolerance bounds marked 1 and -1 indicating +1 standard rate respectively. Figure 10C shows a properly banked turn; figure 10D shows the canonical error of overbanking on entry into the turn.
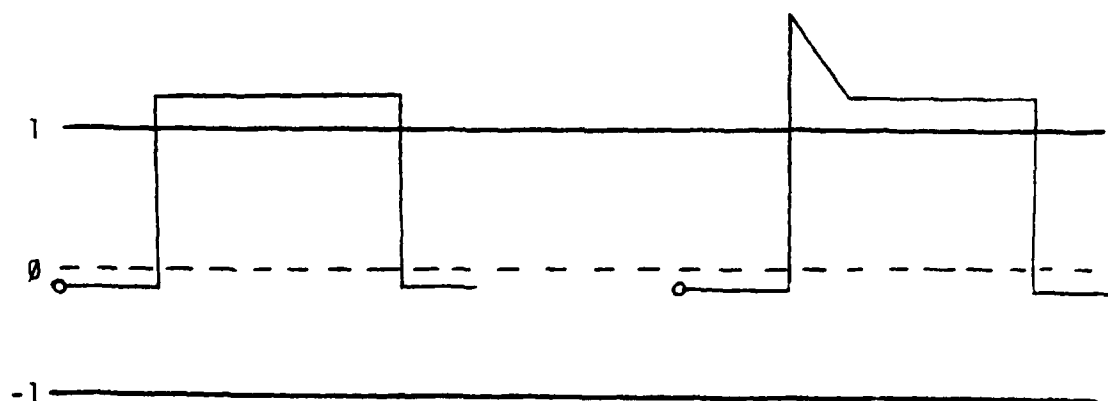
Two other ILS procedures and associated canonical errors are shown in Figure 11. Figures 11A and 11B are diagrams of the altitude parameter in which MDA denotes the minimum descent altitude and the tolerance bounds designate +50 ft. intervals around the MDA. Figure 11A shows the correct procedure of leveling off before reaching the MDA; figure 11B shows a failure to stop descent - it corresponds to the canonical error called "busting your altitude." Figures 11C and 11D are diagrams of the glide slope, GSI. In figure 11C the GSI is kept properly centered; Figure 11D shows the classical canonical error called "chasing the needle."

Some canonical errors jointly effect two or more task parameters. The pattern recognition procedures for detecting these errors are essentially the same as those used in the single parameter cases.

C+5

C

C-5

A. Correct initiation   B. Beginning a turn
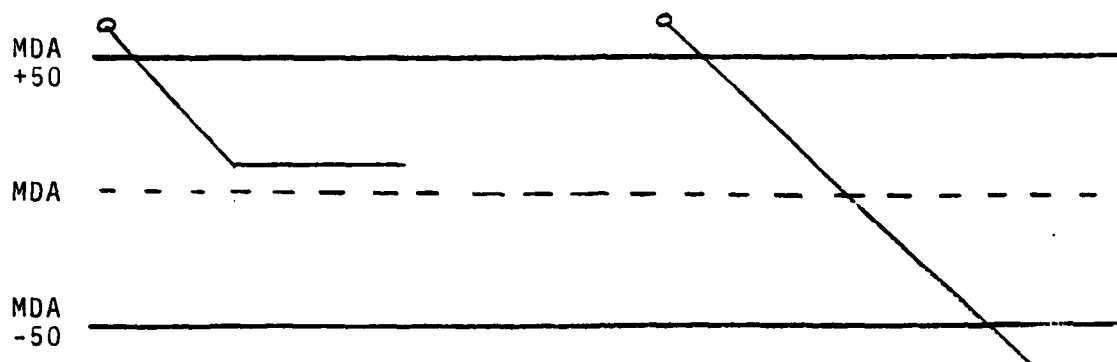   of turn   in wrong direction

1

∅

-1

C. Constant rate turn   D. Overbanking on
                                  turn entry

Figure 10. Correct Turn Maneuvers and Canonical Turn Errors

MDA
+50

MDA

MDA
-50

A. Stopping descent at the   B. Busting Altitude
    Minimum Descent Altitude

GS
+Tol.

GS

GS
-Tol.

C. Proper tracking   D. Chasing the Needle
    of the GSI

Figure 11. Correct ILS Maneuvers and Canonical ILS Errors

SECTION V

CONCLUSIONS

This report documents the work done during the first two years of a research program originally planned as a four-year effort. The major thrust of the work to date has been methodological development and has consisted primarily in extending the performance analysis methods developed in earlier research for data aggregation and linearization, semantic segmentation, and error diagnosis. These methods are designed to yield very specific, clearly articulated descriptions of the salient features of trainee performance at a relatively deep level of detail. They are designed to mirror both the analysis methods used by expert instructors and the results obtained by them.

We have demonstrated that these methods are general and extensible by newly applying them to another task domain, ILS approaches. We have shown that the task-invariant methods are directly applicable to these new tasks. These tasks are a great deal more complex than the original tasks for which the system was initially developed, instrument turn patterns. They involve navigational decision-making skills, as well as flight control skills and the further addition of en-route instrument failures greatly complicates task performance. Also, these complications greatly augment the number and variety of possible errors. Thus, the operation of the major task-specific component of the model, the module responsible for detecting canonical errors, has also been validated in principle through preliminary testing. The data representations used for describing performance and canonical errors appear to be general and the algorithms for pattern recognition driven by these descriptions appear to be effective.

The design for a comprehensive flight training system, including capabilities for task demonstration, real-time (in-flight) analysis, diagnosis of underlying performance difficulties, remediation and task assignment, has been developed. The post-flight performance analysis model developed in this research is the key, central component of this system. We feel that the implementation of the full-scale system is feasible and that the development effort would greatly improve understanding of the learning and teaching of complex skills and greatly extend the state-of-the-art of simulator-based training.

DATE
FILMED

-8